

App. No. 09/939,162
Amendment Dated February 28, 2005
Reply to Office Action of January 26, 2005

Listing of claims:

Claim 1 (Currently amended): A computer-implemented method for providing feedback regarding a program's data access patterns, comprising:

identifying repetitively occurring data access sequences in a stream of data access references;

displaying a plurality of identifiers, wherein each identifier is associated with one of the data access sequences;

upon selection of one of the plurality of identifiers, identifying code related to the data access sequence associated with the selected identifier; and

generating a stream flow output that displays the frequency that repetitively occurring data access sequences follow other repetitively occurring data access sequences when non-repetitively data access sequences are ignored. ~~occurrences of repetitively occurring data access sequences in the stream of data access references while ignoring non-repetitively occurring data access sequences.~~

Claim 2 (original): The method of claim 1, wherein identifying the sequences includes steps, comprising:

constructing a grammar from the stream of data access references;

building a candidate sequence using the grammar; and

if a cost of accessing data in the candidate sequence exceeds a threshold, marking the candidate sequence as a repetitively occurring data access sequence.

Claim 3 (original): The method of claim 2, wherein computing the cost comprises multiplying a number of times the candidate sequence occurs in the grammar by a number of data access references in the candidate sequence.

Claim 4 (original): The method of claim 1, wherein the list of identifiers is displayed in a software development tool.

App. No. 09/939,162
Amendment Dated February 28, 2005
Reply to Office Action of January 26, 2005

Claim 5 (original): The method of claim 4, wherein the software development tool is a visual development environment.

Claim 6 (original): The method of claim 4, wherein the selection is received from a user input device.

Claim 7 (original): The method of claim 1, further comprising displaying a navigation pane that displays the list of identifiers and navigates the list in response to user input.

Claim 8 (original): The method of claim 1, wherein the code is displayed and highlighted.

Claim 9 (original): The method of claim 8, wherein the code comprises source code.

Claim 10 (original): The method of claim 8, wherein the code comprises assembly code.

Claim 11 (original): The method of claim 1, wherein the code, when previously executed, referenced data in the data access sequence associated with the selected identifier.

Claim 12 (original): The method of claim 1, wherein the stream of data access references is included in a trace file.

Claim 13 (Currently amended): A system for developing computer-executable software, comprising:

- an instrumentation tool configured to instrument a software program to produce a trace when the software program is executed;

- a trace analyzer configured to receive the trace and identify repetitively occurring data access sequences;

- a stream flow detector that is configured to generate a stream flow output that displays the frequency that repetitively occurring data access sequences follow other repetitively occurring data access sequences when non-repetitively data access sequences are ignored.

App. No. 09/939,162
Amendment Dated February 28, 2005
Reply to Office Action of January 26, 2005

~~occurrences of repetitively occurring data access sequences in the stream of data access references while ignoring non-repetitively occurring data access sequences;~~ and
a software development tool configured to use the identified data access sequences and stream flow output in software development.

Claim 14 (original): The system of claim 13, wherein the instrumentation tool, the trace analyzer, and the software development tool are integrated in a visual development environment.

Claim 15 (original): The system of claim 13, wherein the trace analyzer identifies repetitively occurring data access sequences by performing steps, comprising:

- constructing a directed acyclic graph (DAG) from the data accesses of the trace file;
- building a candidate sequence using the DAG; and
- if a cost of accessing data in the candidate sequence exceeds a threshold, marking the candidate sequence as a repetitively occurring data access sequence.

Claim 16 (original): The system of claim 15, wherein computing the cost comprises multiplying a number of times the candidate sequence is generated using the DAG by a number of data access references in the candidate sequence.

Claim 17 (original): The system of claim 13, wherein the software development tool includes a navigation pane configured to display a list of identifiers, wherein each identifier is associated with one of the repetitively occurring data access sequences.

Claim 18 (original): The system of claim 17, wherein the navigation pane is further configured to navigate the list of identifiers in response to user input.

Claim 19 (original): The system of claim 18, wherein upon selection of one of the identifiers, the software development tool is further configured to display code related to the repetitively occurring data access sequence associated with the selected identifier.

App. No. 09/939,162
Amendment Dated February 28, 2005
Reply to Office Action of January 26, 2005

Claim 20 (Currently amended): A computer-readable medium having computer-executable instructions for providing information about a stream of data accesses, comprising:
displaying a plurality of identifiers, wherein each identifier is associated with a repetitively occurring data access sequence;

upon selection of one of the identifiers, displaying an attribute of the associated data access sequence; and

generating a stream flow output that displays the frequency that repetitively occurring data access sequences follow other repetitively occurring data access sequences when non-repetitively data access sequences are ignored. ~~occurrences of repetitively occurring data access sequences in the stream of data access references while ignoring non-repetitively occurring data access sequences.~~

Claim 21 (original): The computer-readable medium of claim 20, wherein the attribute comprises the number of times the associated data access sequence repeats in the stream.

Claim 22 (original): The computer-readable medium of claim 20, wherein the attribute comprises the number of data references in the associated data access sequence.

Claim 23 (original): The computer-readable medium of claim 20, wherein the attribute comprises a number of unique objects referenced in the associated data access sequence.

Claim 24 (original): The computer-readable medium of claim 20, wherein the attribute comprises a number of references in the stream between occurrences of the data access sequence.

Claim 25 (Previously submitted): The computer-readable medium of claim 20, wherein the attribute shows an efficiency with which the references of a data access sequence are placed in cache blocks.

Claim 26 (original): The computer-readable medium of claim 25, wherein the efficiency is calculated by dividing a minimum number of cache blocks in which references of the data

App. No. 09/939,162
Amendment Dated February 28, 2005
Reply to Office Action of January 26, 2005

access sequence could be placed by an actual number of cache blocks in which references in the data access sequence are placed.